Universidade de Brasília
Instituto de Ciências Exatas

Departamento de Ciência da Computação

CIC 117536 - Projeto e Análise de Algoritmos

Terceira Prova

Turma: B

NP-completude

Prof. Flávio L. C. de Moura

6 de dezembro de 2018

1. **(2.5 pontos)** O problema 2-SAT tem como instâncias as fórmulas lógicas formadas por conjunções de disjunções de até dois literais, onde um literal é uma variável booleana ou a negação de uma variável booleana. Por exemplo, a expressão a seguir é uma instância de 2-SAT:

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge x_3$$

Prove que 2-SAT $\in$ P.

Solução. Dizer que uma instância de 2-SAT $\in$ P significa encontrar um algoritmo que descubra se existem ou não valores para os literais tais que a fórmula final seja verdadeira, e fazê-lo em tempo polinomial.

Por ser composta de conjunções, a fórmula é dita verdadeira se cada uma de suas cláusulas é verdadeira. Isso significa que $(x_1 \vee \neg x_2)$, $(\neg x_1 \vee$

$\neg x_3$), $(x_1 \vee x_2)$ e x_3, neste caso, devem ser todas verdadeiras.

Para que uma fórmula arbitrária $(A \vee B)$ seja verdadeira, temos que:

(a) Se A = 0, B deve ser 1

(b) Se B = 0, A deve ser 1

Sendo assim, substituímos $(A \vee B)$ por $(\neg A \implies B) \wedge (\neg B \implies A)$ e criamos um grafo de inferências, onde cada literal se torna dois vértices (ele e sua negação) e cada aresta equivale a uma implicação das definidas acima para toda a conjunção.

Com esse grafo, temos diversas novas inferências, pois se $A \implies B$ e $B \implies C$, então $A \implies C$. Isso significa que cada caminho novo entre vértices se torna uma implicação.

Assim, basta avaliarmos as componentes fortemente conexas do grafo construído e verificar se uma variável e sua negação estão ou não presentes em alguma delas. A presença significa dizer que $A \implies \neg A$ e $\neg A \implies A$, um absurdo: não há valores para os literais que tornam a instância verdadeira; e a ausência significa o contrário, que há valores que o façam.

Sendo assim, 2-SAT pode ser feito em tempo polinomial, pois como já visto anteriormente, calcular componentes fortemente conexas pode ser feito em O(V+E), e depois de realizar este passo, basta verificar se um literal e sua negação estão ao mesmo tempo em alguma das SCC com algum algoritmo de busca, também polinomial.

2. **(2.5 pontos)** Em aula, assumimos que SAT é um problema NP-completo (Teorema de Cook-Levin), e a partir deste fato mostramos que 3-SAT e CLIQUE também são problemas NP-completos. As reduções foram feitas de acordo com o seguinte diagrama:

$$\begin{tikzcd} SAT \arrow[d] \\ 3\text{-}SAT \arrow[d] \\ CLIQUE \end{tikzcd}$$

Um ciclo Hamiltoniano é um ciclo simples que visita cada vértice de um grafo exatamente uma vez. Considere o problema de decisão HAM-CYCLE que pergunta se um dado grafo (não-dirigido) G possui um ciclo Hamiltoniano. Mostre que HAM-CYCLE é um problema NP-completo. Sua solução deve ser construída a partir de SAT, 3-SAT ou CLIQUE. Caso, você não veja como reduzir diretamente HAM-CYCLE a partir destes, mas sabe como fazê-lo a partir de um certo problema Q então inicialmente mostre que Q é NP-completo a partir de SAT, 3-SAT ou CLIQUE, e assim por diante. Digamos que você não saiba como mostrar que Q é NP-completo diretamente a partir de SAT, 3-SAT ou CLIQUE, mas você sabe como fazê-lo a partir de outro problema Q', e também sabe como mostrar que Q' é NP-completo a partir de 3-SAT, por exemplo. Então o diagrama correspondente à sua solução seria:

$$
\begin{tikzcd}
SAT \arrow[d] & & & \\
3\text{-}SAT \arrow[r] \arrow[d] & Q' \arrow[r] & Q \arrow[r] & \text{HAM-CYCLE} \\
CLIQUE & & &
\end{tikzcd}
$$

E todas as reduções (de 3-SAT para Q', de Q' para Q e de Q para HAM-CYCLE) devem ser detalhadas na sua solução.

Solução.

Para provar que HAM-CYCLE é um problema NP-completo, precisamos satisfazer duas condições: provar que está em NP e que é NP-difícil.

A primeira parte do problema é facilmente solucionada, tendo em vista que seria possível criar um algoritmo em tempo polinomial que descobre se um grafo contém um ciclo hamiltoniano: basta atribuirmos cores aos vértices (para ter certeza de que só visitamos cada um deles uma única vez) e percorrer todos os possíveis caminhos entre um vértice e ele mesmo, fazendo as devidas alterações para a limitação da visitação única, verificando se em alguma dessas respostas todos os vértices foram visitados (como um caixeiro-viajante para todos os nós).

Para a segunda parte do problema, precisamos converter qualquer instância de um problema conhecido no espaço de NP-completos em um

HAM-CYCLE em tempo polinomial. Vamos provar transformando um 3-SAT em um ciclo hamiltoniano.

Criamos, então, um grafo a partir de uma instância 3-SAT, onde cada literal se torna um vértice e uma cláusula é uma ligação entre vértices. Com isso, montamos as arestas do grafo de forma que a instância 3-SAT é verdadeira se conseguimos percorrer o grafo em uma direção só chegando ao final no vértice inicial. Isso configura um ciclo hamiltoniano e prova que 3-SAT pode ser reduzido a HAM-CYCLE. Como 3-SAT é NP-completo e 3-SAT $\leq_{\mathrm{p}}$ HAM-CYCLE, então HAM-CYCLE é NP-difícil, segundo teorema dado em sala.

3. **(2.5 pontos)** Considere o seguinte jogo em um grafo (não-dirigido) G, que inicialmente contém 0 ou mais bolas de gude em seus vértices: um movimento deste jogo consiste em remover duas bolas de gude de um vértice $v \in G$, e adicionar uma bola a algum vértice adjacente de v. Agora, considere o seguinte problema: Dado um grafo G, e uma função $p(v)$ que retorna o número de bolas de gude no vértice v, existe uma sequência de movimentos que remove todas as bolas de G, exceto uma? Mostre que este problema é NP-completo. A mesma observação feita no exercício anterior vale aqui: a prova deve ser feita a partir de problemas que provamos serem NP-completos, e reduções intermediárias, caso existam, devem ser incluídas na solução.

   Solução. Para provar que este problema é NP-completo, precisamos provar que pertence a NP e é NP-difícil.

   Inicialmente, para provar que pertence a NP, precisamos criar um algoritmo que adivinhe uma solução para o problema e, em seguida, verifique se esta é correta.

   Assim, a partir de uma sequência de jogadas determinada, testamos uma a uma e realizamos a verificação final: num grafo com cada vértice contendo o atributo "número de bolas", retiramos 2 do vértice escolhido inicialmente e adicionamos 1 a um vértice adjacente. Repetimos este passo até o fim da sequência, verificando ao final se restou apenas 1 bola no total dos vértices. Se não houver, refazemos a sequência para outros vértices adjacentes possíveis a serem adicionados. Ao final de testar todas as possibilidades, se a verificação ainda não for satisfatória, aquela sequência não é possível para o jogo, e em qualquer momento que ela for satisfatória, o algoritmo termina porque a sequência é correta.

Por fim, para mostrar que o problema das bolas de gude é NP-difícil, precisamos mostrar que algum NP-completo pode ser reduzido a ele. Isso pode ser mostrado com HAM-CYCLE, já que provamos que este é NP-completo na questão anterior. Para isso, a partir de um HAM-CYCLE, queremos criar um problema de bolas de gude. Temos que se o algoritmo de HAM-CYCLE é rodado em um grafo e a saída é que existe um ciclo hamiltoniano nele, precisamos que exista uma sequência de jogadas que resulte em uma única bola de gude na mesa ao final dela, e vice-versa.

Um ciclo hamiltoniano pode ser descrito como o problema das bolas de gude suficientemente se cada vértice deste último for visitado. Isso ocorre pois sempre formamos um ciclo no problema das bolas de gude: cada jogada consiste basicamente em visitar um nó e ir a seu adjacente. Ao fim da sequência, se todos os vértices contiverem 0 bolas de gude e 1 deles contiver uma, significa que fechamos um ciclo e visitamos cada vértice uma única vez, ou seja, temos um ciclo hamiltoniano.

4. **(2.5 pontos)** Uma fórmula booleana em *forma normal conjuntiva com disjunção exclusiva (FNCX)* é uma conjunção de diversas cláusulas, e cada cláusula é uma disjunção exclusiva (XOR) de diversos literais. Lembre-se que a disjunção exclusiva é dada por:

a	b	$a \oplus b$
V	V	F
V	F	V
F	V	V
F	F	F

O problema FNCX-SAT pergunta se uma dada fórmula em FNCX é satisfatível. Mostre que o problema FNCX-SAT está em P, ou então que FNCX-SAT é NP-completo. No último caso, a mesma observação feita nos dois exercícios anteriores vale aqui: a prova deve ser feita a partir de problemas que provamos serem NP-completos, e reduções intermediárias, caso existam, devem ser incluídas na solução.

Solução. A partir da tabela-verdade da disjunção exclusiva, percebe-se que esta pode ser lida como uma disjunção de conjunções, como a seguir:

$$A \oplus B = A\overline{B} + \overline{A}B$$

Essa transformação pode ser feita em tempo linear, visto que apenas realiza atribuições a cada cláusula, transformando-as em disjunções de variantes dos literais (negados ou não).

Assim, pensamos em $A\overline{B} = A_1$ e $\overline{A}B = A_2$ e a cláusula pode ser vista como $(A_1 \vee A_2)$.

Então, temos que um problema FNCX-SAT reduz para um problema SAT em tempo polinomial, como visto acima. Sendo assim, FNCX-SAT $\in$ P: basta reduzirmos as disjunções exclusivas em disjunções de conjunções, e então rodar o mesmo algoritmo utilizado para problemas SAT.